— Solution notes —

COMPUTER SCIENCE TRIPOS Part IA – 2004 – Paper 1

6 Foundations of Computer Science (ACN)

This question has been translated from Standard ML to OCaml

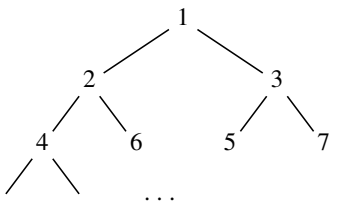
In OCaml it is possible to use functions as values: they can be passed as arguments and returned as results. Explain the notation used to write a function without having to give it a name. [2 marks]

Answer:

```
fun x -> A
```

This question looks at two different ways of implementing functional arrays.

(*a*) One possible functional implementation of an array is based on trees, and the path to a stored value follows the binary code for the subscript:



where in the above diagram the numbers show where in the tree a value with the given subscript will live.

Write code that creates, retrieves values from and updates an array that has this representation, and using big-O notation explain the associated costs. [8 marks]

Answer: I will give here the code to read from the tree-style array, but update is then trivial and follows from it.

```
let rec access n (B(v, l, r)) =
  if n = 1 then
    v
  else if n mod 2 = 1 then
    access ((n - 1) / 2) r
  else
    access (n / 2) l
```

Cost is guaranteed O(log n) where n is subscript being used.

(*b*) A different way of handling functional arrays is to represent the whole array directly by a function that maps from integers to values. To access the item at position k in such an array you just use the array as a function and give it

k as its argument, and to update the array you need to create a new function reflecting the changed value.

If the array is to hold integer values, what OCaml type does it have? [1 mark]

Answer:

```
int -> int
```

Write a function `update a n v` where `a` is a functional array in this style, `n` is an integer index and `v` is a new value. The result of the call to `update` must behave as an array that stores all the values that `a` did except that it is as if an assignment of the style "`a[n] := v`" has been performed. [5 marks]

Answer:

```
let update a n v =
  fun i -> if i = n then v else a n (* easy *)
```

In big-O notation, what is the cost of your `update` function? After a sequence of updates what is the cost of accessing the array? [4 marks]

Answer: Cost of update is O(1), but cost of access is linear in number of updates done.
